Activity 1 – MessageComponent.js

```
import React, { useState }
from "react";

function MessageComponent() {
  const [message, setMessage]
= useState("Hello Student");

  return (
    <div>
      <h2>{message}</h2>
      <button onClick={() =>
setMessage("Welcome to React
State")}>
        Change Message
      </button>
    </div>
  );
}
export default
MessageComponent;
```

Output:

Activity 2 – Counter.js

```
import React, { useState }
from "react";

function Counter() {
```

```
  const [count, setCount] =
useState(0);

  return (
    <div>
      <h2>Count: {count}</h2>
      <button onClick={() =>
setCount(count +
1)}>Increase</button>
      <button onClick={() =>
setCount(count -
1)}>Decrease</button>
    </div>
  );
}
export default Counter;
```

Output:

Activity 3 – TextInput.js

```
import React, { useState }
from "react";

function TextInput() {
  const [text, setText] =
useState("");

  return (
    <div>
      <input
        type="text"
        value={text}
        onChange={(e) =>
setText(e.target.value)}
        placeholder="Enter
your name"
```

```
      />
      <p>You entered:
{text}</p>
    </div>
  );
}
export default TextInput;
```

Output:

```
z4
```

You entered: z4

Activity 4 – UserDetails.js

```
import React, { useState }
from "react";

function UserDetails() {
  const [name, setName] =
useState("");
  const [city, setCity] =
useState("");

  return (
    <div>
      <input
        type="text"
        placeholder="Enter
Name"
        value={name}
        onChange={(e) =>
setName(e.target.value)}
      />

      <input
        type="text"
        placeholder="Enter
City"
        value={city}
        onChange={(e) =>
setCity(e.target.value)}
      />

      <p>Name: {name}</p>
      <p>City: {city}</p>
    </div>
```

```
  );
}
export default UserDetails;
```

Output:

```
z4                    mumbai
```

Name: z4

City: mumbai

Activity 5 – UserDetailsObject.js

```
import React, { useState }
from "react";

function UserDetailsObject()
{
  const [user, setUser] =
useState({
    name: "",
    city: ""
  });

  const handleChange = (e) =>
{
    const { name, value } =
e.target;

    setUser({
      ...user,
      [name]: value
    });
  };

  return (
    <div>
      <input
        type="text"
        name="name"
        placeholder="Enter
Name"
        value={user.name}

onChange={handleChange}
      />

      <input
        type="text"
```

```
        name="city"
        placeholder="Enter
City"
        value={user.city}

onChange={handleChange}
      />

      <p>Name:
{user.name}</p>
      <p>City:
{user.city}</p>
    </div>
  );
}
export default
UserDetailsObject;
```

Output:

ab1	mumbai

Name: ab1

City: mumbai

Activity 6 – EmailValidation.js

```
import React, { useState }
from "react";

function EmailValidation() {
  const [email, setEmail] =
useState("");
  const [message, setMessage]
= useState("");

  const validateEmail = () =>
{
    if (email === "") {
      setMessage("Email field
cannot be empty.");
    } else if
(!email.includes("@")) {
      setMessage("Invalid
email. Must contain '@'.");
    } else {
      setMessage("Email is
valid!");
    }
```

```
  };

  return (
    <div>
      <input
        type="text"
        placeholder="Enter
Email"
        value={email}
        onChange={(e) =>
setEmail(e.target.value)}
      />
      <button
onClick={validateEmail}>Valid
ate</button>
      <p>{message}</p>
    </div>
  );
}
export default
EmailValidation;
```

Output:

abc@gmail.com	Validate

Email is valid!

abcgmail.com	Validate

Invalid email. Must contain '@'.

Pre-Lab Questions (Answers)

1. What is state in React?

State is a built-in object in React used to store data that can change over time. When state changes, the component re-renders automatically.

2. What is the difference between normal variable and state variable?

Normal Variable	State Variable
Does NOT trigger re-render	Triggers re-render
Value resets on re-render	Value persists between renders
Declared normally	Declared using `useState()`

---

3. What is a controlled component?

A controlled component is an input element whose value is controlled by React state using `useState` and updated using `onChange.`

---

4. Why does React re-render when state changes?

React re-renders because state represents dynamic data. When state updates, React updates the Virtual DOM and re-renders the component to reflect new data.

---

5. Why do we use the spread operator in object state?

The spread operator (`...`) is used to copy existing object properties so we can update one property without deleting the others.